

# Lower bounds for testing digraph connectivity with one-pass streaming algorithms

Glencora Borradaile  
Oregon State University

Claire Mathieu  
Brown University

Theresa Migler  
Oregon State University

April 9, 2014

## Abstract

In this note, we show that three graph properties - strong connectivity, acyclicity, and reachability from a vertex  $s$  to all vertices - each require a working memory of  $\Omega(\epsilon m)$  on a graph with  $m$  edges to be determined correctly with probability greater than  $(1 + \epsilon)/2$ .

In the streaming model of computation, the input is given as a sequence, or *stream*, of elements. There is no random access to the elements; the sequence must be scanned in order. The goal is to process the stream using a small amount of working memory. For an overview see [7]. There has been much research devoted to the study of streaming algorithms, most notably the Gödel-prize winning work of Alon, Matias, and Szegedy [2].

For undirected graph problems, there are many lower bounds in the edge streaming model. Henzinger, Raghaven, and Rajagopalan presented a deterministic lower bound of  $\Omega(n)$  for the working memory required for the following undirected graph problems: computing the connected components, vertex-connected components, and testing graph planarity of  $n$ -vertex graphs [6]. Feigenbaum, Kannan, and Zhang show that any exact, deterministic algorithm for computing the diameter of an undirected graph in the Euclidean plane requires  $\Omega(n)$  bits of working memory [4]. Zelke shows that any algorithm that is able to find a minimum cut of an undirected graph requires  $\Omega(m)$  bits of working memory, this remains true even if randomization is allowed [8].

For directed graphs problems, the ones most likely to come up in analyses of the internet, much less is known. Henzinger et al. showed that for any  $0 < \epsilon < 1$ , estimating the size of the transitive closure of a DAG with relative expected error  $\epsilon$  requires  $\Omega(m)$  bits of working memory [6]. Feigenbaum et al. [3] showed that testing reachability from a given vertex  $s$  to another given vertex  $t$  requires  $\Omega(m)$  bits of space, Guruswami and Onak [5] showed that even with  $p$  passes, the problem requires  $\Omega(n^{1+1/(2(p+1))}/p^{20} \log^{3/2} n)$  bits of space to be solvable with probability at least  $9/10$ .

As for upper bounds in undirected graphs, there are one-pass algorithms for connected components,  $k$ -edge and  $k$ -vertex connectivity ( $k \leq 3$ ), and planarity testing that use  $O(n \log n)$  bits of working memory [6]. There is an algorithm that approximates the diameter within  $1 + \epsilon$  using  $O(\frac{1}{\epsilon})$  bits [4]. For upper bounds in directed graphs, there is an

algorithm that computes the exact size of the transitive closure using  $O(m \log n)$  bits of working memory[6].

In this short note, we consider three basic connectivity questions in directed graphs: determining if a graph is strongly connected, determining if a graph is acyclic, and determining if a vertex  $s$  reaches all other vertices. A directed graph  $G = (V, E)$  is said to be *strongly connected* if for every pair of vertices  $u, v \in V$  there is a path from  $u$  to  $v$  and a path from  $v$  to  $u$ . A directed graph  $G = (V, E)$  is said to be *acyclic* if  $G$  contains no cycles. We say that a vertex  $s$  *reaches* a vertex  $v$  if there is a directed path from  $s$  to  $v$ .

We show that, even with randomization, these graph properties each require  $\Omega(m)$  bits of working memory to be decided with probability greater than  $(1 + \epsilon)/2$  by a one-pass streaming algorithm on  $n$  vertices and  $m$  edges. For these lower bounds we will use simple reductions from the index problem (or the bit-vector problem) in communication complexity:

Alice has a bit-vector  $x$  of length  $m$ . Bob has an index  $i \in \{1, 2, \dots, m\}$  and wishes to know the  $i$ th bit of  $x$ . The only communication allowed is from Alice to Bob.

The following is a rewording of Theorem 2 from Ablayev [1].

**Theorem 1** *For Bob to correctly determine  $x_i$  with probability better than  $\frac{(1+\epsilon)}{2}$ ,  $\Omega(\epsilon m)$  bits of communication are required.*

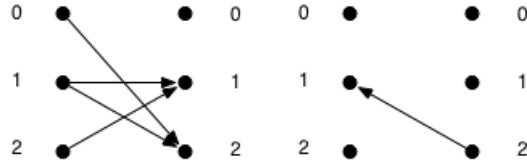
We will now state and prove our main Lemma:

**Lemma 2** *Any algorithm that correctly determines the following graph properties with probability better than  $\frac{(1+\epsilon)}{2}$  requires  $\Omega(\epsilon m)$  bits of working memory: acyclicity, strong connectivity, and reachability of all from  $s$ .*

**Proof:** We reduce from the index problem and use Theorem 1. Let  $x$  denote the  $m$ -bit vector owned by Alice. We define the stream using two sets of edges  $E_1, E_2$ . The edge stream first has the edges of  $E_1$  in arbitrary order, followed by the edges of  $E_2$ , also in arbitrary order. The set  $E_1$  is entirely determined by the  $m$ -bit vector  $x$  owned by Alice, and the set  $E_2$  is entirely determined by the index  $i$  owned by Bob. The graph defined by  $E_1 \cup E_2$  has  $\Omega(\sqrt{m})$  vertices, and  $E_1$  has  $O(m)$  edges. To solve the index problem, Alice constructs  $E_1$  and simulates the streaming algorithm up to the point when  $E_1$  has arrived, then sends to Bob the current state of the memory. Upon reception of the message, Bob constructs  $E_2$  and continues the simulation up to the point when  $E_2$  has finished arriving. Bob's final decision is then determined by the outcome of the streaming algorithm. Thus, the lemma will be proved.

**Acyclicity** Let  $n = \lceil \sqrt{m} \rceil$  and let  $V = L \cup R$ , where  $L$  and  $R$  both have size  $n$  and have vertices labeled 0 through  $n - 1$ .  $E_1$  is the bipartite graph that has an edge from vertex  $j \in L$  to vertex  $k \in R$  iff  $x$  has a 1 in position  $jn + k$ .  $E_2$  consists of a single edge determined by Bob's bit  $i$ : let  $k = i \bmod n$  and  $j = \frac{i-k}{n}$ . Then  $E_2$  consists of the edge from vertex  $k \in R$  to vertex  $i \in L$ . See figures 1(a) and 1(b) for an illustration of an example  $E_1$  and  $E_2$ .

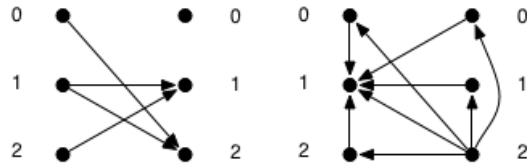
Observe that  $E_1 \cup E_2$  is acyclic iff  $x_i = 0$ , thus the reduction is complete.



(a) An example of  $E_1$  (b) An example of  $E_2$  corresponding to the bit-responding to  $B$ 's index vector 001011010. being 5,  $j = 1$  and  $k = 2$ .

**Strong connectivity** The construction for  $E_1$  is the same as in the acyclic case.  $E_2$  consists of  $4n - 2$  edges determined by Bob's bit  $i$ : let  $k = i \bmod n$  and  $j = \frac{i-k}{n}$ . Then  $E_2$  consists of all edges from  $k$  to  $V - \{k\}$ , and from  $V - \{j\}$  to  $j$ . See figures 1(c) and 1(d) for an illustration of an example  $E_1$  and  $E_2$ .

We claim that  $G$  is strongly connected iff  $x_i = 1$ . Indeed, if  $G$  is strongly connected, there must be a path from  $j$  to  $k$ . The only edges leaving  $j$  are to vertices in  $R$  and the only edges entering  $k$  are from vertices in  $L$ . And the only edges extending from  $R$  to  $L$  are either entering  $j$  or leaving  $k$ . Thus, the only possible path from  $j$  to  $k$  is the single edge from  $j$  to  $k$  which is present only when  $x_i = 1$ . Now suppose that  $x_i = 1$ .  $k$  can certainly reach every vertex and every vertex can reach  $j$ . Since the edge from  $j$  to  $k$  is present, we know that every vertex can reach  $k$  and  $k$  can reach every vertex. Therefore,  $G$  is strongly connected. Thus the reduction is complete.

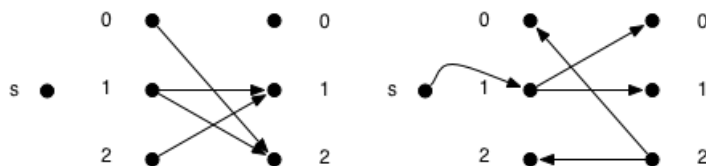


(c) An example of  $E_1$  (d) An example of  $E_2$  corresponding to the bit-responding to  $B$ 's index vector 001011010. being 5,  $j = 1$  and  $k = 2$ .

**Reachability from  $s$**   $E_1$  is as above with additional vertex  $s$  with in and out degree 0.

$E_2$  consists of  $2n - 1$  edges determined by Bob's bit  $i$ : let  $k = i \bmod n$  and  $j = \frac{i-k}{n}$ . Then  $E_2$  consists of one edge from  $s$  to  $j \in L$ ,  $n - 1$  edges from  $j \in L$  to  $R - \{k\}$ , and  $n - 1$  edges from  $k \in R$  to  $L - \{j\}$ . See figures 1(e) and 1(f) for an illustration of an example  $E_1$  and  $E_2$ .

We claim that in  $G$   $s$  reaches everything iff  $x_i = 1$ . Indeed, if  $s$  can reach all vertices in  $G$ , and the only edge from  $s$  is to  $j$ ,  $j$  must be able to reach all vertices in  $G - \{s\}$ . In particular  $j$  must reach  $k$ . The only edges extending from  $R$  to  $L$  are from  $k$ , so the only way for  $j$  to reach  $k$  is by the edge from  $j$  to  $k$  which is present only when  $x_i = 1$ . Now suppose  $x_i = 1$ . We know  $s$  reaches  $j$  and therefore all of  $R$ , including  $k$ , and  $k$  reaches all of  $L - \{j\}$ . Therefore,  $s$  reaches all vertices of  $G$ . Thus the reduction is complete.



(e) An example of  $E_1$  corresponding to the bit-vector 001011010. (f) An example of  $E_2$  corresponding to  $B$ 's index being 5,  $j = 1$  and  $k = 2$ .

□

## References

- [1] Farid Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theoretical Computer Science*, 157:139–159, 1996.
- [2] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 20–29, New York, NY, USA, 1996. ACM.
- [3] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2):207–216, December 2005.
- [4] Joan Feigenbaum, Sampath Kannan, and Jian Zhang. Computing diameter in the streaming and sliding-window models. *Algorithmica*, pages 25–41, 2004.
- [5] Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. *CoRR*, abs/1212.6925, 2012.
- [6] Monika R. Henzinger, Prabhakar Raghavan, and Sridhar Rajagopalan. Computing on data streams. In *External memory algorithms*, pages 107–118. American Mathematical Society, 1999.

- [7] S. Muthukrishnan. *Data Streams: Algorithms and Applications*. Now Publishers Inc, 1 edition, 2005.
- [8] Mariano Zelke. Intractability of min- and max-cut in streaming graphs. *Inf. Process. Lett.*, 111(3):145–150, jan 2011.